

PPPPPPPPPPPPP AAAAAAAA SSSSSSSSSSS RRRRRRRRRRRR TTTTTTTTTTTTTLLL
PPPPPPPPPPPPP AAAAAAAA SSSSSSSSSSS RRRRRRRRRRRR TTTTTTTTTTTTTLLL
PPPPPPPPPPPPP AAAAAAAA SSSSSSSSSSS RRRRRRRRRRRR TTTTTTTTTTTTTLLL
PPP PPP AAA AAA SSS RRR RRR TTT LLL
PPP PPP AAA AAA SSS RRR RRR TTT LLL
PPP PPP AAA AAA SSS RRR RRR TTT LLL
PPP PPP AAA AAA SSS RRR RRR TTT LLL
PPP PPP AAA AAA SSS RRR RRR TTT LLL
PPP PPP AAA AAA SSS RRR RRR TTT LLL
PPP PPP AAA AAA SSS RRR RRR TTT LLL
PPPPPPPPPPPPP AAA AAA SSSSSSSSS RRRRRRRRRRRR TTT LLL
PPPPPPPPPPPPP AAA AAA SSSSSSSSS RRRRRRRRRRRR TTT LLL
PPPPPPPPPPPPP AAA AAA SSSSSSSSS RRRRRRRRRRRR TTT LLL
PPP AAAAAAAAAAAA SSS RRR RRR TTT LLL
PPP AAAAAAAAAAAA SSS RRR RRR TTT LLL
PPP AAAAAAAAAAAA SSS RRR RRR TTT LLL
PPP AAA AAA SSSSSSSSSSS RRR RRR TTT LLL
PPP AAA AAA SSSSSSSSSSS RRR RRR TTT LLL
PPP AAA AAA SSSSSSSSSSS RRR RRR TTT LLL

FILEID**PASCLOSE2

6 14

PPPPPPPP P AAAAAA SSSSSSSS CCCCCCCC LL 000000 SSSSSSSS EEEEEEEE 222222
PPPPPPPP P AAAAAA SSSSSSSS CCCCCCCC LL 000000 SSSSSSSS EEEEEEEE 222222
PP PP AA AA SS CC LL 00 00 SS EE 22 22
PP PP AA AA SS CC LL 00 00 SS EE 22 22
PP PP AA AA SS CC LL 00 00 SS EE 22 22
PPPPPPPP AA AA SSSSSS CC LL 00 00 SSSSSS EEEEEEE 22 22
PPPPPPPP AA AA SSSSSS CC LL 00 00 SSSSSS EEEEEEE 22 22
PP AAAAAAAAAA SS CC LL 00 00 SS EE 22 22
PP AAAAAAAAAA SS CC LL 00 00 SS EE 22 22
PP AA AA SS CC LL 00 00 SS EE 22 22
PP AA AA SS CC LL 00 00 SS EE 22 22
PP AA AA SSSSSSSS CCCCCCCC LLLLLLLL 000000 SSSSSSSS EEEEEEEE 2222222222
PP AA AA SSSSSSSS CCCCCCCC LLLLLLLL 000000 SSSSSSSS EEEEEEEE 2222222222

LL IIIII SSSSSSSS
LL IIIII SSSSSSSS
LL II SS
LL II SS
LL II SS
LL III SSSSSS
LL III SSSSSS
LL II SS
LL II SS
LL II SS
LLLLLLL LLLLIII SSSSSSSS
LLLLLLL LLLLIII SSSSSSSS

PA
1-

```
1 0001 0 MODULE PASSCLOSE2 ( %TITLE 'Close a file'  
2 0002 0 IDENT = '1-006' ! File: PASCLOSE2.B32 Edit: SBL1006  
3 0003 0 ) =  
4 0004 1 BEGIN  
5 0005 1 *****  
6 0006 1 *  
7 0007 1 *  
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY  
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.  
10 0010 1 * ALL RIGHTS RESERVED.  
11 0011 1 *  
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
17 0017 1 * TRANSFERRED.  
18 0018 1 *  
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
21 0021 1 * CORPORATION.  
22 0022 1 *  
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.  
25 0025 1 *  
26 0026 1 *  
27 0027 1 *****  
28 0028 1 :  
29 0029 1 :  
30 0030 1 ++  
31 0031 1 : FACILITY: Pascal Language Support  
32 0032 1 : ABSTRACT:  
33 0033 1 :  
34 0034 1 : This module contains PASSCLOSE2 and PASS$CLOSE, which close a file.  
35 0035 1 : ENVIRONMENT: User mode - AST reentrant  
36 0036 1 :  
37 0037 1 : AUTHOR: Steven B. Lionel, CREATION DATE: 1-April-1981  
38 0038 1 :  
39 0039 1 : MODIFIED BY:  
40 0040 1 :  
41 0041 1 :  
42 0042 1 : 1-001 - Original. SBL 1-April-1981  
43 0043 1 : 1-002 - Give error in PASSCLOSE2 if FCB is not valid. SBL 21-June-1982  
44 0044 1 : 1-003 - Correct erroneous error check in PASS$CLOSE. SBL 9-July-1982  
45 0045 1 : 1-004 - Make sure XAB chain doesn't point to garbage. SBL 29-July-1982  
46 0046 1 : 1-005 - Zero FCBSL STATUS before closing file. QAR FT3-2. SBL 30-Aug-1982  
47 0047 1 : 1-006 - Use PASS$AB KEYWD_NAME_TABLE to detect duplicate keywords.  
48 0048 1 :  
49 0049 1 : SBL 19-Aug-1983  
50 0050 1 :--  
51 0051 1 :--
```

```
: 53      0052 1 %SBTTL 'Declarations'  
:  
: 54      0053 1  
:  
: 55      0054 1 PROLOGUE DEFINITIONS:  
:  
: 56      0055 1  
:  
: 57      0056 1  
:  
: 58      0057 1 REQUIRE 'RTLIN:PASPROLOG';           ! Linkages, externals, PSECTs, structures  
:  
: 59      0121 1  
:  
: 60      0122 1  
:  
: 61      0123 1 TABLE OF CONTENTS:  
:  
: 62      0124 1  
:  
: 63      0125 1  
:  
: 64      0126 1 FORWARD ROUTINE  
:  
: 65      0127 1      PASSCLOSE2: NOVALUE,          ! Called by compiled code  
:  
: 66      0128 1      PASS$CLOSE: NOVALUE;        ! Called internally by RTL  
:  
: 67      0129 1  
:  
: 68      0130 1  
:  
: 69      0131 1 MACROS:  
:  
: 70      0132 1      NONE  
:  
: 71      0133 1  
:  
: 72      0134 1  
:  
: 73      0135 1 EQUATED SYMBOLS:  
:  
: 74      0136 1      NONE  
:  
: 75      0137 1  
:  
: 76      0138 1  
:  
: 77      0139 1 FIELDS:  
:  
: 78      0140 1      NONE  
:  
: 79      0141 1  
:  
: 80      0142 1  
:  
: 81      0143 1 OWN STORAGE:  
:  
: 82      0144 1  
:  
: 83      0145 1      NONE
```

```

85      0146 1 %SBTTL 'PASSCLOSE2 - Close a file'
86      0147 1 GLOBAL ROUTINE PASSCLOSE2 (
87          0148 1 PFV: REF $PAS$PFV FILE VARIABLE,
88          0149 1 KEYWORDS: VECTOR [, LONG]
89          0150 1 ): NOVALUE =
90      0151 1
91      0152 1 ++
92      0153 1 | FUNCTIONAL DESCRIPTION:
93      0154 1 |
94          0155 1 | This routine closes a Pascal file. This entry is called from the
95          0156 1 | compiled code for the CLOSE procedure.
96      0157 1
97      0158 1 | CALLING SEQUENCE:
98      0159 1 |
99      0160 1 | CALL PASSCLOSE2 (PFV.mr.r [, keyword.rz.v [, keyword.rz.v ...]])
100     0161 1
101     0162 1 | FORMAL PARAMETERS:
102     0163 1
103     0164 1 | PFV           - The Pascal File Variable (PFV) passed by reference.
104     0165 1 |             The structure of the PFV is defined in PASPFV.REQ.
105     0166 1
106     0167 1 | keyword        - Zero or more values representing keywords in the
107     0168 1 |             CLOSE procedure. Some keywords are followed by
108     0169 1 |             associated arguments and values. See PASOPEKEY.REQ
109     0170 1 |             for the keyword value definitions.
110     0171 1
111     0172 1 | If the keyword ERROR is specified, it must be first
112     0173 1 |             in the list, otherwise it will be ignored.
113     0174 1
114     0175 1 | IMPLICIT INPUTS:
115     0176 1 |             NONE
116     0177 1
117     0178 1 | IMPLICIT OUTPUTS:
118     0179 1 |             NONE
119     0180 1
120     0181 1 | COMPLETION STATUS: (or ROUTINE VALUE:)
121     0182 1 |             NONE
122     0183 1
123     0184 1 | SIDE EFFECTS:
124     0185 1 |             See PASSCLOSE
125     0186 1
126     0187 1 | SIGNALLED ERRORS:
127     0188 1 |             FILALRCLO - file already closed
128     0189 1
129     0190 1
130     0191 1
131     0192 1
132     0193 1
133     0194 1
134     0195 1 |--|
135     0196 1
136     0197 2 | BEGIN
137     0198 2
138     0199 2 | LOCAL
139         0200 2 | FCB: REF $PAS$FCB_CONTROL_BLOCK,
140         0201 2 | PFV ADDR: VOLATILE,
141         0202 2 | UNWIND_ACT: VOLATILE,

```

! File Control Block
! Enable argument
! Enable argument

```
142      0203 2      ERROR_ADDR: VOLATILE;           ! Enable argument
143      0204 2
144      0205 2      BUILTIN
145      0206 2      ACTUALCOUNT,
146      0207 2      AP,
147      0208 2      CALLG;
148      0209 2
149      0210 2
150      0211 2      ENABLE
151      0212 2      PASS$IO_HANDLER (PFV_ADDR, UNWIND_ACT, ERROR_ADDR);
152      0213 2
153      0214 2      |+ Set PFV_ADDR enable argument.
154      0215 2      |-_
155      0216 2
156      0217 2      PFV_ADDR = PFV [PFV$R_PFV];
157      0218 2
158      0219 2      |+
159      0220 2      | Validate and lock PFV.
160      0221 2      |-_
161      0222 2
162      0223 2      PASS$VALIDATE_PFV (PFV [PFV$R_PFV]; FCB);
163      0224 2
164      0225 2
165      0226 2      |+
166      0227 2      | Set unwind action to unlock file.
167      0228 2      |-_
168      0229 2      UNWIND_ACT = PAS$K_UNWIND_UNLOCK;
169      0230 2
170      0231 2
171      0232 2      |+
172      0233 2      | If ERROR is present, it is the first keyword in the list. See
173      0234 2      | if that is true. If so, set enable argument to the return address.
174      0235 2      |-_
175      0236 2      IF ACTUALCOUNT () GEQU 3    ! At least 3 needed for PFV, keyword, value
176      0237 2      THEN
177      0238 2      |+ IF .KEYWORDS [0] EQL PAS$K_ERROR          ! Is it the ERROR keyword?
178      0239 2      |  THEN
179      0240 2      |    ERROR_ADDR = .KEYWORDS [1];
180      0241 2
181      0242 2
182      0243 2      |+
183      0244 2      | If file variable is not valid, call PASS$OPEN_IMPLICIT to see if the
184      0245 2      | file should be implicitly opened. Even though we will just close it
185      0246 2      | again, we must do this to satisfy the language semantics for INPUT
186      0247 2      | and OUTPUT.
187      0248 2      |-_
188      0249 2      IF NOT .PFV [PFV$V_VALID]
189      0250 2      THEN
190      0251 2      |+ PASS$OPEN_IMPLICIT (PFV [PFV$R_PFV], FCB [FCBSR_FCB]; FCB);
191      0252 2
192      0253 2
193      0254 2      |+
194      0255 2      | If FCB pointer is not valid, give an error.
195      0256 2      |-_
196      0257 2      IF NOT .PFV [PFV$V_FCB_VALID]
197      0258 2      THEN
198      0259 2      |+ SPASS$IO_ERROR (PASS$_FILALRCLO,0);
```

```

199      0260 2
200      0261 2
201      0262 2
202      0263 2
203      0264 2
204      0265 2
205      0266 2
206      0267 2
207      0268 2
208      0269 2
209      0270 2
210      0271 2
211      0272 2
212      0273 2
213      0274 2
214      0275 2
215      0276 2
216      0277 2
217      0278 2
218      0279 2
219      0280 2
220      0281 2
221      0282 2
222      0283 2
223      0284 2
224      0285 2
225      0286 2
226      0287 2
227      0288 2
228      0289 1

        0260 2
        0261 2
        0262 2
        0263 2
        0264 2
        0265 2
        0266 2
        0267 2
        0268 2
        0269 2
        0270 2
        0271 2
        0272 2
        0273 2
        0274 2
        0275 2
        0276 2
        0277 2
        0278 2
        0279 2
        0280 2
        0281 2
        0282 2
        0283 2
        0284 2
        0285 2
        0286 2
        0287 2
        0288 2
        0289 1

```

+ Zero the STATUS value in the FCB. This will get stored in the PFV by PASS\$REMOVE_FILE.

- FCB = .PFV [PFV\$A_FCB];
FCB [FCBSL_STATUS] = 0;

+ Call PASS\$CLOSE to do the work.

- CALLG (.AP, PASS\$CLOSE);

+ Remove it from the list of open files.

- PASS\$REMOVE_FILE (FCB [FCBSR_FCB]);

+ Unlock the file variable.

- PFV [PFV\$V_LOCK] = 0;

RETURN;

END:

! End of routine PASSCLOSE2

.TITLE PASSCLOSE2 Close a file
.IDENT \1-006\

.EXTRN PASSCLOSE2, PASS\$CLOSE
.EXTRN PASS\$IO_HANDLER
.EXTRN PASS\$VALIDATE_PFV
.EXTRN PASS\$OPEN_IMPPLICIT
.EXTRN PASS\$SIGNAL, PASSK_FINALRCLO
.EXTRN PASS\$REMOVE_FILE

.PSECT _PASS\$CODE,NOWRT, SHR, PIC,2

5E 6D 08 04 005E 56 AE 00000000G 04 AE 03 19	00C4 00000 08 C2 00002 7E D4 00005 AE 7C 00007 CF DE 0000A AC D0 0000F 56 D0 00013 00 16 00017 01 D0 0001D 6C 91 00021 0A 1F 00024 AC D1 00026 04 12 0002A	ENTRY SUBL2 CLRL CLRQ MOVAL MOVL MOVL JSB MOVL CMPB BLSSU CMPL BNEQ	PASSCLOSE2, Save R2,R6,R7 #8, SP ERROR ADDR UNWIND ACT 4\$, (FP) PFV, R6 R6, PFV ADDR PASS\$VALIDATE_PFV #1, UNWIND_ACT (AP), #3 1\$ KEYWORDS, #25 15	: 0147 : 0197 : 0217 : 0223 : 0229 : 0236 : 0239 :
---	--	---	---	---

PASSCLOSE2
1-006

Close a file
PASSCLOSE2 - Close a file

M 14
16-Sep-1984 01:29:07
14-Sep-1984 12:51:21
VAX-11 Bliss-32 V4.0-742
[PASRTL.SRC]PASCLOSE2.B32;1

Page 6
(3)

		6E	0C	AC	DO	0002C	1\$:	MOVL	KEYWORDS+4, ERROR_ADDR	: 0240	
		06	06	A6	EB	00030		BLBS	6(R6), 2\$: 0249	
		OE	07	A6	00	16	00034	JSB	PASS\$OPEN_IMPLICIT	: 0251	
					06	E0	0003A	BBS	#6, 7(R6), 3\$: 0257	
					7E	D4	0003F	CLRL	-(SP)	: 0259	
					00000000G	00	8F	9A	#PASS\$K_FILALRCLO, -(SP)		
						02	FB	00045	CALLS	#2, PASS\$SIGNAL	
							04	0004C	RET		
						57	OC	A6	MOVL	12(R6), FCB	: 0266
						D4	A7	D4	CLRL	-44(FCB)	: 0267
						0000V	CF	6C	CALLG	(AP), PASS\$CLOSE	: 0273
								FA	PUSHL	FCB	: 0279
						00000000G	00	57	CALLS	#1, PASS\$REMOVE_FILE	
							DD	00059	MOVL	PFV_R0	: 0285
							01	FB	BIORB2	#128, 7(R0)	
						50	AC	DO	RET		: 0289
						07	A0	00062	.WORD	Save nothing	: 0197
							80	8A	MOVL	8(AP), R0	
								00066	MOVL	4(R0), R0	
								04	PUSHAB	ERROR_ADDR	
								0006B	PUSHAB	UNWIND_ACT	
								0000	PUSHAB	PFV_ADDR	
								0006C	PUSHL	#3	
								4\$:	PUSHL	SP	
									MOVQ	4(AP), -(SP)	
									CALLS	#3, PASS\$IO_HANDLER	
									RET		

: Routine Size: 143 bytes. Routine Base: _PASS\$CODE + 0000

: 229 0290 1
: 230 0291 1 !<BLF/PAGE>

P/
1-

```
: 232      0292 1 %SBTTL 'PASSCLOSE - Close a file'  
: 233      0293 1 GLOBAL ROUTINE PASSCLOSE (  
: 234          IN PFV: REF SPASSPFV FILE_VARIABLE.  
: 235          KEYWORDS: VECTOR [, [ONG]]  
: 236          ): NOVALUE =  
:  
: 237          ! Close a file  
: 238          ! File variable  
: 239          ! Keywords and specifiers  
:  
: 240          ++  
: 241          FUNCTIONAL DESCRIPTION:  
:  
: 242          This procedure closes the Pascal file specified by "PFV"  
: 243          using the attributes possibly supplied as "keyword". PASSCLOSE  
: 244          is to be called from other RTL routines only after they have tested  
: 245          and set PFV$V_LOCK in "PFV".  
:  
: 246          CALLING SEQUENCE:  
:  
: 247          PASSCLOSE (PFV.mr.r [, keyword.rz.v [, keyword.rz.v ...]])  
:  
: 248          FORMAL PARAMETERS:  
:  
: 249          PFV           - The Pascal File Variable (PFV) passed by reference.  
: 250          The structure of the PFV is defined in PASPFV.MDL.  
:  
: 251          keyword        - Zero or more values representing keywords in the  
: 252          CLOSE procedure. Some keywords are followed by  
: 253          associated arguments and values. See PASOPNKEY.MDL  
: 254          for the keyword value definitions.  
:  
: 255          IMPLICIT INPUTS:  
:  
: 256          The PFV$V_LOCK bit is assumed to have been tested-and-set by the caller.  
: 257          The PFV$V_FCB_VALID bit is assumed set.  
:  
: 258          IMPLICIT OUTPUTS:  
:  
: 259          NONE  
:  
: 260          COMPLETION STATUS:  
:  
: 261          NONE  
:  
: 262          SIDE EFFECTS:  
:  
: 263          The file is closed. The record buffer and resultant name string  
: 264          are deallocated.  
:  
: 265          SIGNALLED ERRORS:  
:  
: 266          ERRDURCLO - error during CLOSE  
: 267          INVARGPAS - invalid argument to Pascal Run-Time Library  
: 268          FILNAMREQ - file name required to save, submit or print file  
:  
: 269          --  
:  
: 270          BEGIN  
:  
: 271          LOCAL
```

```

: 289      0349 2      PFV: REF SPASSPFV_FILE_VARIABLE,      ! Pascal File Variable
: 290      0350 2      FCB: REF SPASSFCB_CONTROL_BLOCK,    ! File control block
: 291      0351 2      BLOCK_ADDR: VOLATILE,          ! Address of dynamically allocated block
: 292      0352 2      STATUS,                      ! Status from $CLOSE
: 293      0353 2      KEYWORDS_SEEN: BITVECTOR [PASSK_OPEKEYHI+1], ! Keep track of keywords seen
: 294      0354 2      USER_ACTION_BPV: REF VECTOR [, LONG]; ! USER_ACTION routine BPV
: 295      0355 2
: 296      0356 2
: 297      0357 2      BUILTIN
: 298      0358 2      TESTBITCS,
: 299      0359 2      TESTBITSC,
: 300      0360 2      ACTUALCOUNT;
: 301      0361 2
: 302      0362 2      BIND
: 303      0363 2      RAB = FCB: REF BLOCK [, BYTE],
: 304      0364 2      FAB = FCB: REF SPASSFAB_FCB_STRUCT;
: 305      0365 2
: 306      0366 2
: 307      0367 2      !+ Get PFV address.
: 308      0368 2      !-
: 309      0369 2
: 310      0370 2      PFV = .IN_PFV;
: 311      0371 2
: 312      0372 2      !+
: 313      0373 2      ! Get addresses of PFV.
: 314      0374 2      !-
: 315      0375 2
: 316      0376 2      FCB = .PFV [PFVSA_FCB];
: 317      0377 2
: 318      0378 2      !+
: 319      0379 2      ! Do the keyword processing.
: 320      0380 2      !-
: 321      0381 2
: 322      0382 2      KEYWORDS_SEEN <0,32> = 0; ! Indicate no keywords seen
: 323      0383 2      KEYWORDS_SEEN <32,PASSK_OPEKEYHI+31> = 0;
: 324      0384 2
: 325      0385 2
: 326      0386 2
: 327      0387 2      !+
: 328      0388 2      ! Look through keyword list and set all specified attributes.
: 329      0389 2      !-
: 330      0390 2      INCR I FROM 0 TO (ACTUALCOUNT () - 2) DO
: 331      0391 2      BEGIN
: 332      0392 2
: 333      0393 2      LOCAL
: 334      0394 2      KEYWD_VALUE;
: 335      0395 2
: 336      0396 2
: 337      0397 2      !+
: 338      0398 2      ! Check for valid keyword code. If not already seen, use the
: 339      0399 2      keyword.
: 340      0400 2
: 341      0401 2      KEYWD_VALUE = .KEYWORDS [.I];
: 342      0402 2      IF .KEYWD_VALUE LSS PASSK_OPEKEYLO OR .KEYWD_VALUE GTR PASSK_OPEKEYHI
: 343      0403 2      THEN
: 344      0404 2      $PASSIO_ERROR (PASS_INVARGPAS,0);
: 345      0405 2

```

```
: 346      0406 3      IF TESTBITS (KEYWORDS_SEEN [.PASS$AB_KEYWD_NAME_TABLE[.KEYWD_VALUE]])  
: 347      0407 3      THEN  
: 348      0408 3      CASE .KEYWD_VALUE FROM PASSK_OPEKEYLO TO PASSK_OPEKEYHI OF  
: 349      0409 3      SET  
: 350      0410 3  
: 351      0411 3      [PASSK_DISPOSITION_SAVE]:  
: 352      0412 4      BEGIN  
: 353      0413 4      FCB [FCBSV_DELETE] = 0;  
: 354      0414 4      FCB [FCBSV_PRINT] = 0;  
: 355      0415 4      FCB [FCBSV_SUBMIT] = 0;  
: 356      0416 4      FCB [FCBSV_SAVE] = 1;  
: 357      0417 3      END;  
: 358      0418 3  
: 359      0419 3      [PASSK_DISPOSITION_DELETE]:  
: 360      0420 4      BEGIN  
: 361      0421 4      FCB [FCBSV_PRINT] = 0;  
: 362      0422 4      FCB [FCBSV_SUBMIT] = 0;  
: 363      0423 4      FCB [FCBSV_SAVE] = 0;  
: 364      0424 4      FCB [FCBSV_DELETE] = 1;  
: 365      0425 3      END;  
: 366      0426 3  
: 367      0427 3      [PASSK_DISPOSITION_PRINT]:  
: 368      0428 4      BEGIN  
: 369      0429 4      FCB [FCBSV_DELETE] = 0;  
: 370      0430 4      FCB [FCBSV_SUBMIT] = 0;  
: 371      0431 4      FCB [FCBSV_SAVE] = 1;  
: 372      0432 4      FCB [FCBSV_PRINT] = 1;  
: 373      0433 3      END;  
: 374      0434 3  
: 375      0435 3      [PASSK_DISPOSITION_PRINT_DELETE]:  
: 376      0436 4      BEGIN  
: 377      0437 4      FCB [FCBSV_SUBMIT] = 0;  
: 378      0438 4      FCB [FCBSV_SAVE] = 0;  
: 379      0439 4      FCB [FCBSV_PRINT] = 1;  
: 380      0440 4      FCB [FCBSV_DELETE] = 1;  
: 381      0441 3      END;  
: 382      0442 3  
: 383      0443 3      [PASSK_DISPOSITION_SUBMIT]:  
: 384      0444 4      BEGIN  
: 385      0445 4      FCB [FCBSV_PRINT] = 0;  
: 386      0446 4      FCB [FCBSV_DELETE] = 0;  
: 387      0447 4      FCB [FCBSV_SUBMIT] = 1;  
: 388      0448 4      FCB [FCBSV_SAVE] = 1;  
: 389      0449 3      END;  
: 390      0450 3  
: 391      0451 3      [PASSK_DISPOSITION_SUBMIT_DELETE]:  
: 392      0452 4      BEGIN  
: 393      0453 4      FCB [FCBSV_PRINT] = 0;  
: 394      0454 4      FCB [FCBSV_SAVE] = 0;  
: 395      0455 4      FCB [FCBSV_SUBMIT] = 1;  
: 396      0456 4      FCB [FCBSV_DELETE] = 1;  
: 397      0457 3      END;  
: 398      0458 3  
: 399      0459 3      [PASSK_USER_ACTION]:  
: 400      0460 4      BEGIN  
: 401      0461 4      USER_ACTION_BPV = .KEYWORDS [(I=.I+1)];  
: 402      0462 3      END;
```

```
: 403      0463 3
: 404      0464 3
: 405      0465 4
: 406      0466 4
: 407      0467 4
: 408      0468 4
: 409      0469 4
: 410      0470 4
: 411      0471 3
: 412      0472 3
: 413      0473 3
: 414      0474 3
: 415      0475 3
: 416      0476 3
: 417      0477 3
: 418      0478 2
: 419      0479 2
: 420      0480 2
: 421      0481 2
: 422      0482 2
: 423      0483 2
: 424      0484 2
: 425      0485 3
: 426      0486 2
: 427      0487 2
: 428      0488 2
: 429      0489 2
: 430      0490 2
: 431      0491 2
: 432      0492 2
: 433      0493 2
: 434      0494 2
: 435      0495 2
: 436      0496 2
: 437      0497 2
: 438      0498 2
: 439      0499 2
: 440      0500 2
: 441      0501 2
: 442      0502 2
: 443      0503 2
: 444      0504 2
: 445      0505 2
: 446      0506 2
: 447      0507 2
: 448      0508 2
: 449      0509 2
: 450      0510 2
: 451      0511 3
: 452      0512 2
: 453      0513 2
: 454      0514 2
: 455      0515 2
: 456      0516 2
: 457      0517 2
: 458      0518 2
: 459      0519 2

        [PASS$K_ERROR]:
        BEGIN
        !+
        | The ERROR parameter, if specified, was already processed
        | in PASSCLOSE, so just ignore it here.
        !-
        | = .I + 1; ! Ignore next parameter
        END;

        [INRANGE_OUTRANGE]:
        $PAS$IO_ERROR (PASS_INVARGPAS,0);

        TES;

        END;

        !+
        | Check for incompatible DISPOSITION.
        !-

        IF .FAB [FAB$V_TMD] AND
           (.FCB [FCBSV_SAVE] OR .FCB [FCBSV_SUBMIT] OR .FCB [FCBSV_PRINT])
        THEN
          $PAS$IO_ERROR (PASS_FILENOREQ,0);      ! File name required

        !+
        | Fill in fields in RMS control blocks
        !-

        IF .FCB [FCBSV_DELETE] AND NOT .FAB [FAB$V_TMD]
        THEN
          FAB [FAB$V_DLT] = 1;

        IF .FCB [FCBSV_SUBMIT]
        THEN
          FAB [FAB$V_SCF] = 1;

        IF .FCB [FCBSV_PRINT]
        THEN
          FAB [FAB$V_SPL] = 1;

        !+
        | If a textfile, and if in Generation mode, and if the buffer
        | is not empty, do a WRITELN.
        !-

        IF .FCB [FCBSV_TEXT] AND .FCB [FCBSV_GENERATION] AND
           (.FCB [FCBSA_RECORD_CUR] GTRA .FCB [FCBSA_RECORD_BEG])
        THEN
          PASS$WRITELN (PFV [PFV$R_PFV], FCB [FCBSR_FCB]);

        !+
        | Make sure the XAB chain doesn't point to garbage.
        !-

        FAB [FAB$L_XAB] = 0;
```

```
460      0520 2
461      0521 2
462      0522 2  |+ Close the file.
463      0523 2
464      0524 2
465      0525 2  IF .FAB [FABSW_IFI] NEQ 0
466      0526 2
467      0527 3  THEN BEGIN
468      0528 3  IF .KEYWORDS_SEEN [PASSK_USER_ACTION]
469      0529 3  THEN BEGIN
470      0530 4  LINKAGE
471      0531 4    USER_ACTION_LNK = CALL (REGISTER=1,STANDARD,STANDARD);
472      0532 4
473      0533 4
474      0534 4  STATUS = USER_ACTION_LNK (.USER_ACTION_BPV [0],           | Address
475      0535 4          .USER_ACTION_BPV [1],           | Environment
476      0536 4          FAB [0,0,0,0],           | FAB address
477      0537 4          RAB [0,0,0,0],           | RAB address
478      0538 4          PFV [PFV$R_PFV]);           | PFV address
479      0539 4  END
480      0540 4
481      0541 4  ELSE STATUS = $PASS$RMS_OP ($CLOSE (FAB=FAB [0,0,0,0]));
482      0542 4
483      0543 4
484      0544 4  |+ Set the IFI to zero so we don't try another $CLOSE if the last one
485      0545 4  failed.
486      0546 4
487      0547 4
488      0548 4
489      0549 4
490      0550 4
491      0551 4
492      0552 4
493      0553 4
494      0554 4
495      0555 4
496      0556 4  IF NOT .STATUS
497      0557 4  THEN $PASS$IO_ERROR (PASS_ERRDURCLO);
498      0558 2
499      0559 2
500      0560 2
501      0561 2  |+ Deallocate the record buffer, if necessary.
502      0562 2
503      0563 2
504      0564 2
505      0565 2
506      0566 2  IF TESTBITS_C (FCB [FCBSV_DYNAMIC_UBF])
507      0567 2  THEN PASS$FREE_VM (.RAB [RABSW_USZ], RAB [RABSL_UBF]);
508      0568 2
509      0569 2  |+ Deallocate key types vector, if necessary.
510      0570 2
511      0571 2
512      0572 2  IF .FCB [FCBSA_KEY_TYPES] NEQ 0
513      0573 2  THEN PASS$FREE_VM (.FCB [FCBSL_NKEYS]*4, FCB [FCBSA_KEY_TYPES]);
514      0574 2
515      0575 2
516      0576 2  |+ Deallocate resultant name string, if dynamic. This should be the
```

PAS\$CLOSE2
1-006Close a file
PAS\$CLOSE - Close a fileF 15
16-Sep-1984 01:29:07
14-Sep-1984 12:51:21VAX-11 Bliss-32 V4.0-742
[PASRTL.SRC]PASCLOSE2.B32;1Page 12
(4)

```

517 0577 2 ! last thing to do.
518 0578 2
519 0579 2
520 0580 2 IF TESTBITS(FCB [FCBSV_DYNAMIC_RSN])
521 0581 2 THEN
522 0582 2 BEGIN
523 0583 2 LOCAL
524 0584 2 NAM: REF BLOCK [, BYTE];
525 0585 2 NAM = .FAB [FABSL_NAM];
526 0586 2 PAS$FREE_VM (.NAM [NAM$B_RSL], NAM [NAM$L_RSA]);
527 0587 2 END;
528 0588 2
529 0589 2
530 0590 2 !+ Mark PFV to indicate file closed.
531 0591 2
532 0592 2
533 0593 2 PFV [PFVSV_OPEN] = 0;
534 0594 2 PFV [PFVSV_VALID] = 0;      ! Invalidate flags
535 0595 2 PFV [PFVSV_DFB] = 0;      ! Invalidate file buffer
536 0596 2 PFV [PFVSV_EOF_DEFINED] = 0;
537 0597 2
538 0598 2 RETURN;
539 0599 2
540 0600 1 END:           ! End of routine PAS$CLOSE

```

```

.EXTRN PASSK_INVARGPAS
.EXTRN PAS$AB_KEYWD_NAME_TABLE
.EXTRN PASSK_FILNAMREQ
.EXTRN PAS$WITELN, SYSSCLOSE
.EXTRN PASSK_ERRDURCLO
.EXTRN PAS$FREE_VM

```

		OFFC 00000			
			.ENTRY	PASS\$CLOSE, Save R2,R3,R4,R5,R6,R7,R8,R9,-	0293
			MOVAB	PASS\$SIGNAL, R11	
			MOVAB	PASS\$FREE_VM, R10	
			SUBL2	#8, SP	
			MOVL	IN PFV, PFV	0370
			MOVL	12(PFV), FCB	0376
			CLRL	KEYWORD\$ SEEN	0382
			BICW2	#511, KEYWORDS_SEEN+4	0383
			MOVZBL	(AP), R4	0390
			SUBL2	#2, R4	
			MNEGL	#1 I	
			BRW	15\$	
			MOVL	KEYWORDS[I], KEYWD_VALUE	0401
			BLEQ	4\$	0402
			CMPL	KEYWD_VALUE, #40	
			BGTR	4\$	
			MOVZBL	PASS\$AB_KEYWD_NAME_TABLE[KEYWD_VALUE], R0	0406
			BBSS	R0, KEYWORDS_SEEN-1\$	
			CASEL	KEYWD_VALUE, #1, #39	0408
0050	0050	0050	WORD	4\$-3\$,-	
0050	0050	0050		4\$-3\$,-	
0050	0050	0050		4\$-3\$,-	

PASSCLOSE2
1-006

CLOSE a file
PASS\$CLOSE - Close a file

G 15
16-Sep-1984 01:29:07 VAX-11 Bliss-32 v4.0-742
14-Sep-1984 12:51:21 [PASRTL.SRC]PASCLOSE2.B32;1

Page 13
(4)

7E	00G	7E	D4	0009B	4\$:	CLRL	-(SP)	
		8F	9A	0009D		MOVZBL	#PASSK_INVARGPAS, -(SP)	
		73	11	000A1		BRB	17\$	
FC	A7	01C0	8F	AA	000A3	5\$:	BICW2	#448, -4(FCB)
			2B	11	000A9		BRB	10\$
FC	A7	01A0	8F	AA	000AB	6\$:	BICW2	#416, -4(FCB)
			32	11	000B1		BRB	12\$
FC	A7	0140	8F	AA	000B3	7\$:	BICW2	#320, -4(FCB)
FC	A7	A0	8F	88	000B9		BISB2	#160, -4(FCB)
			37	11	000BE		BRB	15\$
FC	A7	0120	8F	AA	000C0	8\$:	BICW2	#288, -4(FCB)
FC	A7	80	8F	88	000C6		BISB2	#128, -4(FCB)
			18	11	000CB		BRB	12\$
FC	A7	C0	8F	8A	000CD	9\$:	BICB2	#192, -4(FCB)
FD	A7		01	88	000D2		BISB2	#1, -3(FCB)
FC	A7		20	88	000D6	10\$:	BISB2	#32, -4(FCB)
			1B	11	000DA		BRB	15\$
FC	A7	A0	8F	8A	000DC	11\$:	BICB2	#160, -4(FCB)
FD	A7	01	88	000E1		BISB2	#1, -3(FCB)	
FC	A7	40	8F	88	000E5	12\$:	BISB2	#64, -4(FCB)

PAS\$CLOSE2
1-006Close a file
PAS\$\$CLOSE - Close a fileH 15
16-Sep-1984 01:29:07
14-Sep-1984 12:51:21 VAX-11 Bliss-32 V4.0-742
[PASRTL.SRC]PASCLOSE2.B32;1Page 14
(4)

			OB	11	000EA	BRB	15\$	0408
			52	D6	000EC	INCL	I	0461
		59	08	AC42	D0 000EE	MOVL	KEYWORDS[I], USER_ACTION_BPV	
			02	11	000F3	BRB	15\$	
			52	D6	000F5	INCL	I	0408
			54	F1	000F7	ACBL	R4, #1, I, 2\$	0470
			04	E1	000FD	BBC	#4, 72(FAB), 18\$	0390
			05	E0	00102	BBS	#5, -4(FAB), 16\$	0484
			FD	A7	E8 00107	BLBS	-3(FAB), 16\$	0485
			FC	A7	95 0010B	TSTB	-4(FAB)	
			0A	18	0010E	BGEQ	18\$	
			7E	D4	00110	CLRL	-(SP)	0487
			00G	8F	9A 00112	MOVZBL	#PASSK FILNAMREQ, -(SP)	
			68	02	FB 00116	CALLS	#2, PAS\$\$SIGNAL	
			04	00119		RET		
			0A	58	F C A7 9E 0011A	MOVAB	-4(FAB), R8	0493
			68	06	E1 0011E	BBC	#6, (R8), 19\$	
			48	A7	04 E0 00122	BBS	#4, 72(FAB), 19\$	
			49	A7	80 88 00127	BISB2	#128, 73(FAB)	0495
			05	01	A8 E9 0012C	BLBC	1(R8), 20\$	0497
			49	A7	40 88 00130	BISB2	#64, 73(FAB)	0499
					68 95 00135	TSTB	(R8)	0501
					04 18 00137	BGEQ	21\$	
			49	A7	20 88 00139	BISB2	#32, 73(FAB)	0503
			11	F8	A7 E9 0013D	BLBC	-8(FAB), 22\$	0510
			68	0C	E1 00141	BBC	#12, (R8), 22\$	
			E8	A7	EC A7 D1 00145	CMPL	-20(FAB), -24(FAB)	0511
					06 1B 0014A	BLEQU	22\$	
					00000000G 00 16 0014C	JSB	PASS\$WRITELN	0513
					68 A7 D4 00152	CLRL	104(FAB)	0519
					46 A7 B5 00155	TSTW	70(FAB)	0525
					43 13 00158	BEQL	25\$	
			10	03	52 44 A7 9E 0015A	MOVAB	68(R7), R2	0536
					02 E1 0015E	BBC	#2, KEYWORDS_SEEN+3, 23\$	0528
					56 DD 00163	PUSHL	PFV	0538
					8F BB 00165	PUSHR	#^M<R2,R7>	
				00	51 04 A9 D0 00169	MOVL	4(USER ACTION BPV), R1	
					03 FB 0016D	CALLS	#3, 80(USER_ACTION_BPV)	
					19 11 00171	BRB	24\$	
					52 DD 00173	PUSHL	R2	0541
					01 FB 00175	CALLS	#1, SY\$CLOSE	
					50 E8 0017C	BLBS	SS\$STATUS, 24\$	
					50 D1 0017F	CMPL	SS\$STATUS, #98906	
					04 12 00186	BNEQ	24\$	
				E7	03 A8 E8 00188	BLBS	3(R8), 23\$	
				51	50 D0 0018C	MOVL	SS\$STATUS, STATUS	
				46	A7 B4 0018F	CLRW	70(FAB)	0548
				08	51 E8 00192	BLBS	STATUS, 25\$	0554
				7E	00G 8F 9A 00195	MOVZBL	#PASSK ERRDURCLO, -(SP)	0556
				6B	01 FB 00199	CALLS	#1, PAS\$\$SIGNAL	
					04 0019C	RET		
			OA	68	13 E5 0019D	BBCC	#19, (R8), 26\$	0564
					24 A7 9F 001A1	PUSHAB	36(FAB)	0566
				7E	20 A7 3C 001A4	MOVZWL	32(FAB), -(SP)	
				6A	02 FB 001A8	CALLS	#2, PAS\$\$FREE_VM	
					CC A7 D5 001AB	TSTL	-52(FAB)	0571
					08 13 001AE	BEQL	27\$	

P/
1-

PASSCLOSE2
1-006

Close a file
PASS\$CLOSE - Close a file

I 15
16-Sep-1984 01:29:07
14-Sep-1984 12:51:21 VAX-11 Bliss-32 V4.0-742
[PASRTL.SRC]PASCLOSE2.B32;1

Page 15
(4)

7E	DO	A7	CC	A7	9F	001B0	PUSHAB	-52(FAB)	: 0573	
		6A		02	78	001B3	ASHL	#2, -48(FAB), -(SP)		
OE		68		02	FB	001B8	CALLS	#2 PASS\$FREE VM	: 0580	
		50		12	E5	001BB	BBCC	#18, (R8), 28\$: 0585	
		7E		04	A0	9F	001C3	MOVL	108(FAB), NAM	: 0586
		6A		03	A0	9A	001C6	PUSHAB	4(NAM)	
06	A6		2007	8F	AA	001CD	MOVZBL	3(NAM), -(SP)		
				02	FB	001CA	CALLS	#2 PA\$SSFREE-VM	: 0596	
				04	001D3		BICW2	#8199, 6(PFV)		
							RET		: 0600	

; Routine Size: 468 bytes, Routine Base: _PASS\$CODE + 008F

: 541 0601 1
: 542 0602 1 !<BLF/PAGE>

P/
1-

PASSCLOSE2
1-006 Close a file
 PASS\$CLOSE - Close a file

544 0603 1 END
545 0604 1
546 0605 0 ELUDOM

J 15
16-Sep-1984 01:29:07 VAX-11 Bliss-32 v4.0-742
14-Sep-1984 12:51:21 [PASRTL.SRC]PASCLOSE2.B32;1

Page 16
(5)

: ! End of module PASS\$CLOSE2

PSECT SUMMARY

Name	Bytes	Attributes
_PASS\$CODE	611	NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

File	Total	Symbols	Pages Mapped	Processing Time
	Loaded	Percent		
\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	16	0	00:01.0
\$255\$DUA28:[PASRTL.OBJ]PASLIB.L32;1	427	111	25	00:00.4

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LISS:PASCLOSE2/OBJ=OBJ\$:PASCLOSE2 MSRC\$:PASCLOSE2/UPDATE=(ENH\$:PASCLOSE2)

Size: 611 code + 0 data bytes
Run Time: 00:14.1
Elapsed Time: 00:51.1
Lines/CPU Min: 2578
Lexemes/CPU-Min: 19367
Memory Used: 193 pages
Compilation Complete

P
1

0293 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

PASRT0
LIS

PASMACROS
REQ

PASRT1
LIS

PASPROLOG
REQ

PASRT2
LIS

PASOPEDER
REQ

PASRT3
LIS

PASABS1
LIS PASCARD2
LIS

PASRT4
LIS

PASCLOSE2
LIS

PASRTL

PASBUGOD
REQ

PASRT2
LIS PASRT3
LIS

PASEXTERN
REQ

PASKDB
REQ

PASPFD
REQ

PASRT4
LIS

PASRT0
MAP

PASLIB
REQ

PASPFU
REQ

PASBIN
LIS

PASRT5
LIS

PASRT6
LIS

PASLIB
REQ

PASPFU
REQ

PASBIN
LIS

PASRT7
LIS

PASRT8
LIS

PASLIB
REQ

PASPFU
REQ

PASBIN
LIS

PASRT9
LIS

PASRT10
LIS

PASLIB
REQ

PASPFU
REQ

PASBIN
LIS

PASRT11
LIS

PASRT12
LIS

PASLIB
REQ

PASPFU
REQ

PASBIN
LIS

PASRT13
LIS

PASRT14
LIS

PASLIB
REQ

PASPFU
REQ

PASBIN
LIS

PASRT15
LIS

PASRT16
LIS

PASLIB
REQ

PASPFU
REQ

PASBIN
LIS

PASRT17
LIS

PASRT18
LIS

PASLIB
REQ

PASPFU
REQ

PASBIN
LIS

PASRT19
LIS

PASRT20
LIS

PASLIB
REQ

PASPFU
REQ

PASBIN
LIS

PASRT21
LIS

PASRT22
LIS

PASLIB
REQ

PASPFU
REQ

PASBIN
LIS

PASRT23
LIS

PASRT24
LIS

PASLIB
REQ

PASPFU
REQ

PASBIN
LIS

PASRT25
LIS

PASRT26
LIS

PASLIB
REQ

PASPFU
REQ

PASBIN
LIS

PASRT27
LIS

PASRT28
LIS

PASLIB
REQ

PASPFU
REQ

PASBIN
LIS

PASRT29
LIS

PASRT30
LIS

PASLIB
REQ

PASPFU
REQ

PASBIN
LIS

PASRT31
LIS

PASRT32
LIS

PASLIB
REQ

PASPFU
REQ

PASBIN
LIS

PASRT33
LIS

PASRT34
LIS

PASLIB
REQ

PASPFU
REQ

PASBIN
LIS

PASRT35
LIS

PASRT36
LIS

PASLIB
REQ

PASPFU
REQ

PASBIN
LIS

PASRT37
LIS

PASRT38
LIS

PASLIB
REQ

PASPFU
REQ

PASBIN
LIS

PASRT39
LIS

PASRT40
LIS

PASLIB
REQ

PASPFU
REQ

PASBIN
LIS

PASRT41
LIS

PASRT42
LIS

PASLIB
REQ

PASPFU
REQ

PASBIN
LIS

PASRT43
LIS

PASRT44
LIS

PASLIB
REQ

PASPFU
REQ

PASBIN
LIS

PASRT45
LIS

PASRT46
LIS

PASLIB
REQ

PASPFU
REQ

PASBIN
LIS

PASRT47
LIS

PASRT48
LIS

PASLIB
REQ

PASPFU
REQ

PASBIN
LIS

PASRT49
LIS

PASRT50
LIS

PASLIB
REQ

PASPFU
REQ

PASBIN
LIS

PASRT51
LIS

PASRT52
LIS

PASLIB
REQ

PASPFU
REQ

PASBIN
LIS

PASRT53
LIS

PASRT54
LIS

PASLIB
REQ

PASPFU
REQ

PASBIN
LIS

PASRT55
LIS

PASRT56
LIS

PASLIB
REQ

PASPFU
REQ

PASBIN
LIS

PASRT57
LIS

PASRT58
LIS

PASLIB
REQ

PASPFU
REQ

PASBIN
LIS

PASRT59
LIS

PASRT60
LIS

PASLIB
REQ

PASPFU
REQ

PASBIN
LIS

PASRT61
LIS

PASRT62
LIS

PASLIB
REQ

PASPFU
REQ

PASBIN
LIS

PASRT63
LIS

PASRT64
LIS

PASLIB
REQ

PASPFU
REQ

PASBIN
LIS

PASRT65
LIS

PASRT66
LIS

PASLIB
REQ

PASPFU
REQ

PASBIN
LIS

PASRT67
LIS

PASRT68
LIS

PASLIB
REQ

PASPFU
REQ

PASBIN
LIS

PASRT69
LIS

PASRT70
LIS

PASLIB
REQ

PASPFU
REQ

PASBIN
LIS